

# Efficient Estimation of the number of neighbours in Probabilistic K Nearest Neighbour Classification

Ji Won Yoon<sup>a</sup>, Nial Friel<sup>b</sup>

<sup>a</sup>*Center for information security technology (CIST), Korea University, Korea*

<sup>b</sup>*School of Mathematical Sciences, University College Dublin, Ireland*

---

## Abstract

Probabilistic k-nearest neighbour (PKNN) classification has been introduced to improve the performance of original k-nearest neighbour (KNN) classification algorithm by explicitly modelling uncertainty in the classification of each feature vector. However, an issue common to both KNN and PKNN is to select the optimal number of neighbours,  $k$ . The contribution of this paper is to incorporate the uncertainty in  $k$  into the decision making, and in so doing use Bayesian model averaging to provide improved classification. Indeed the problem of assessing the uncertainty in  $k$  can be viewed as one of statistical model selection which is one of the most important technical issues in the statistics and machine learning domain. In this paper, a new functional approximation algorithm is proposed to reconstruct the density of the model (order) without relying on time consuming Monte Carlo simulations. In addition, this algorithm avoids cross validation by adopting Bayesian framework. The performance of this algorithm yielded very good performance on several real experimental datasets.

**Keywords:** Bayesian Inference, Model Averaging, K-free model order estimation

---

## 1. Introduction

Supervised classification is a very well studied problem in the machine learning and statistics literature, where the  $k$ -nearest neighbour algorithm (KNN) is one of the most popular approaches. It amounts to assigning an unlabelled class to the most common class label among  $k$  neighbouring feature vectors. One of the key issues in implementing this algorithm is choosing the number of neighbours  $k$ , and various flavours of cross validation are used for this purpose. However a

drawback to kNN is that it does not have a probabilistic interpretation, for example, no uncertainty is associated with the inferred class label.

There have been several recent papers which addressed this deficiency, [1, 2, 3, 4]. Indeed from such a Bayesian perspective the issue of choosing the value of  $k$  can be viewed as a model (order) selection problem. To date, there exist several different approaches to tackle the model selection problem. One of the most popular approaches is based on *information criteria* including the Akaike Information Criterion (AIC), the Schwarz’s Bayesian Information Criterion (BIC) and the Deviance Information Criterion (DIC) [5, 6, 7]. Given a particular model  $\mathcal{M}_k$ , the well-known AIC and BIC are defined by  $AIC(\mathcal{M}_k) = -2 \log L(\mathcal{M}_k) + 2e(\mathcal{M}_k)$  and  $BIC(\mathcal{M}_k) = -2 \log L(\mathcal{M}_k) + e(\mathcal{M}_k) \log N$  for  $N$  observations where  $L(\mathcal{M}_k)$  and  $e(\mathcal{M}_k)$  denote the likelihood and the number of parameters of  $\mathcal{M}_k$ , respectively.

It is known that many fast functional approximations or *information criterion* techniques do not adequately approximate the underlying posterior distribution of the model order. Furthermore, Monte carlo based estimators can provide approximate distributions of the model order, but typically require excessive computation time.

Our main contribution is to propose a new functional approximation technique to infer the posterior distribution of the model order,  $p(K|\mathcal{Y})$  where  $K$  and  $\mathcal{Y}$  denote the model order and observations, respectively. In particular, this paper demonstrates the applicability of the proposed algorithm by addressing the problem of finding the number of neighbours  $k$  for probabilistic k-Nearest Neighbour (PKNN) classification. In addition, we designed a new symmetrized neighbouring structure for the KNN classifier in order to conduct a fair comparison. From an application point of view, we also classified several benchmark datasets and a few real experimental datasets using the proposed algorithms.

In addition to model selection, we also consider improvements of the KNN approach itself for the purpose of a fair comparison. Although conventional KNN based on euclidean distance is widely used in many application domains, the conventional KNN is not a correct model in that it does not guarantee the symmetry of the neighbouring structure.

It is important to state that PKNN formally defines a Markov random field over the joint distribution of the class labels. In turn this yields a complication from an inferential point of view, since it is well understood that the Markov random field corresponding to likelihood of the class labels involves an intractable normalising constant, sometimes called the partition function in statistical physics, rendering exact calculation of the likelihood function almost always impossible.

Inference for such complicated likelihoods function is an active field of research. In the context of PKNN [1] and [3] use the pseudo-likelihood function [8] as an approximation to the true likelihood. While [2] and [4] consider improvements to pseudolikelihood by using a Monte Carlo auxiliary variable technique, the exchange algorithm, [9] which targets the posterior distribution which involves the true intractable likelihood function. Bayesian model selection is generally a computationally demanding exercise, particularly in the current context, due to the intractability of the likelihood function, and for this reason we use a pseudolikelihood approximation throughout this paper, although our efforts are focused on efficient means to improve upon this aspect using composite likelihood approximations [10].

This paper consists of several sections. Section 3 includes the background of the statistical approaches used in this paper. This section shows two main techniques, k-Nearest Neighbour (KNN) classification and Integrated Laplace Approximation (INLA). For the extended literature review, probabilistic kNN (PKNN) is explained with details. The proposed algorithm is introduced in the section 4. In this section, we introduce a generic algorithm to reconstruct and approximate the underlying model order posterior  $p(K|\mathcal{Y})$  and to efficiently search for the optimal model order  $K^*$ . Afterwards, this section includes how to apply the generic algorithm into PKNN. In section 5, PKNN adopting the proposed algorithms have applied to several real datasets. Finally, we conclude this paper with some discussion of sections 6 and 7.

## 2. Related Work

One of the main aims of this paper is to explore nearest neighbour classification from a model selection perspective. Some popular model selection approaches in the literature include the following. Grenander et al. [11, 12] proposed a model selection algorithm which is based on jump-diffusion dynamics with the essential feature that at random times the process jumps between parameter spaces in different models and different dimensions. Similarly, Markov birth-death processes and point processes can be considered. One of the most popular approaches to infer the posterior distribution and to explore model uncertainty is Reversible Jump Markov Chain Monte Carlo developed by Richardson and Green [13]. The composite model approach of Carlin and Chib [14] is a further approach. The relationships between the issue of choice of pseudo-prior in the case of Carlin and Chib’s product composite model and the choice of proposal densities in the case of reversible jump are discussed by Godsill [15].

In addition, there are a lot of similarities in the clustering domain. For instance, many clustering algorithms such as K-means algorithms, Gaussian Mixture Model (GMM), and Spectral clustering have also the challenging difficulty to infer the number of clusters  $K$  as similarly shown in the estimation of the number of neighbours  $K$  of the (P)KNN.

### 3. Statistical Background

#### 3.1. $k$ -Nearest Neighbour ( $k$ NN) model

In pattern recognition, the  $k$ -Nearest Neighbour algorithm ( $k$ NN) is one of the most well-known and useful non-parametric methods for classifying and clustering objects based on classified features which are close, in some sense, in the feature space. The  $k$ NN is designed with the concept that labels or classes are determined by a majority vote of its neighbours. However, along with such a simple implementation, the  $k$ NN has a sensitivity problem from the locality which are generated from two difficult problems: estimating the decision boundary to determine the boundary complexity and the number of neighbours to be voted. In order to address this problem, adaptive  $k$ NN is proposed to efficiently and effectively calculate the number of neighbours and the boundary [16, 17, 18, 19]. In addition, the probabilistic  $k$ NN (PKNN) model which is more robust than the conventional  $k$ NN has been introduced and developed by Markov chain Monte carlo to estimate the number of neighbours [1, 3]. In this paper, we use the PKNN model since it provides proper likelihood term given a particular model with  $k$  neighbours.

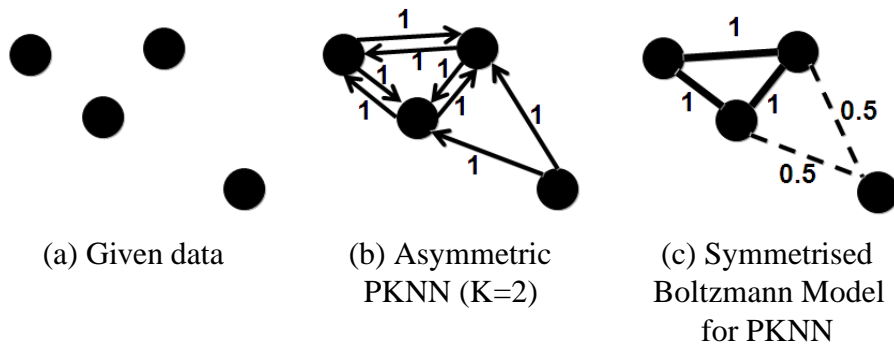


Figure 1: Topological Explanation of PKNN

### 3.1.1. An asymmetric Pseudo-likelihood of PKNN

Let  $\{(z_1, \mathbf{y}_1), (z_2, \mathbf{y}_2), \dots, (z_N, \mathbf{y}_N)\}$  where each  $z_i \in \{1, 2, \dots, C\}$  denote the class label and  $d$  dimensional feature vector  $\mathbf{y}_i \in \mathbb{R}^d$ . Then, the pseudo-likelihood of the probabilistic kNN (PKNN) proposed by [1] can be formed as

$$p(\mathbf{z}|\mathbf{y}, \beta, K) \approx \prod_{i=1}^N \frac{\exp \left\{ \frac{\beta}{K} \sum_{j \in ne(i)} \delta_{z_i, z_j} \right\}}{\sum_{c \in \mathbf{C}} \exp \left\{ \frac{\beta}{K} \sum_{j \in ne(i)} \delta_{c, z_j} \right\}} \quad (1)$$

where the unknown scaling value  $\beta > 0$  and  $\mathbf{C}$  is a set of classes,  $K$  denotes the number of neighbours and  $\delta_{a,b} = 1$  if  $a = b$  and 0 otherwise. In this equation,  $ne(\cdot)$  represents the set of neighbours.

Suppose that we have four data points as shown in Fig. 1-(a). Given  $K = 2$ , we have an interesting network structure in Fig. 1-(b) from this conventional PKNN. In this subgraph, arrows direct the neighbours. As we can see in the Fig. 1-(b), some pairs of data points (nodes) are bidirectional but others are unidirectional, resulting in an asymmetric phenomena. Unfortunately, this asymmetric property does not satisfy the Markov Random Field assumption which can be implicitly applied in Eq. (1).

### 3.1.2. A symmetrised Boltzmann modelling for pseudo-likelihood of PKNN

Since the pseudo-likelihood of the conventional probabilistic kNN is not symmetrised an approximate symmetrised model has been proposed for PKNN [20] as

$$p(\mathbf{z}|\mathbf{y}, \beta, K) \approx \prod_{i=1}^N \frac{\exp \left\{ \frac{\beta}{K} (\sum_{j \in ne(i)} \delta_{z_i, z_j} + \sum_{i \in ne(k)} \delta_{z_i, z_k}) \right\}}{\sum_{c \in \mathbf{C}} \exp \left\{ \frac{\beta}{K} (\sum_{j \in ne(i)} \delta_{c, z_j} + \sum_{i \in ne(k)} \delta_{z_i, z_k}) \right\}}. \quad (2)$$

The Boltzmann modeling of PKNN resolves the asymmetric problem which arises from the conventional PKNN of Eq. (1). However, the Boltzmann modeling reconstructs the symmetrised network by averaging the asymmetrised effects from the principal structure of PKNN as shown in Fig. 1-(c). This brings different interaction rate among the edges. In the subgraph, two edges have a value of a half and all others have a value of one and so this difference may yield an inaccurate Markov Random Field model again.

### 3.2. Estimation of PKNN by Markov chain Monte Carlo (MCMC) - a conventional way

The most popular approach to estimate the parameters of PKNN is using Markov chain Monte Carlo (MCMC). In this paper, PKNN via MCMC is also used for performance comparison. In particular, there are two different version of MCMC.

The first approach is to infer the unknown model parameters ( $\tilde{\beta}$  and  $\tilde{K}$ ) in the training step via MCMC. Afterward, given these estimate values, we can classify the new data from the testing set straightforwardly using the conditional posterior  $p(z_i|\mathbf{y}, \mathbf{z}, \mathbf{y}', \tilde{\beta}, \tilde{K})$ . Suppose that we need to reconstruct the target posterior  $p(\beta, K|\mathbf{z}, \mathbf{y})$  given the observations  $\mathbf{z}$  and  $\mathbf{y}$  which is a set of training data. The standard MCMC approach uses a Metropolis-Hasting (MH) algorithm, so that each unknown parameter is updated according to an acceptance probability

$$\mathcal{A} = \min \left\{ 1, \frac{p(\mathbf{z}|\mathbf{y}, \hat{\beta}, \hat{K})p(\hat{\beta})p(\hat{K})q(\beta, K)}{p(\mathbf{z}|\mathbf{y}, \beta, K)p(\beta)p(K)q(\hat{\beta}, \hat{K})} \right\}$$

where  $\hat{\beta}$  and  $\hat{K}$  denote the proposed new parameters. In the training step, we estimate  $\tilde{\beta}$  and  $\tilde{K}$  from the above MCMC simulation. Afterwards, we simply classify the testing datasets given  $\tilde{\beta}$  and  $\tilde{K}$ . That is, given a testing set we can estimate the classes by

$$z'^* = \arg_{z'} \max p(z'|\mathbf{y}, \mathbf{z}, \mathbf{y}', \tilde{\beta}, \tilde{K})$$

for a new test data  $\mathbf{y}'$  and its unknown label  $z'$ . However, since the uncertainty of the model parameters is ignored in the testing step of the first approach, the first approach with two separate steps (training and testing) is less preferred from a statistical point of view although it is often used in practice. Unlike the first approach, the second approach jointly estimates the hidden model parameters to incorporate this uncertainty while classifying the testing datasets. In the second approach we reconstruct not the conditional distribution  $p(z'|\mathbf{y}, \mathbf{z}, \mathbf{y}', \tilde{\beta}, \tilde{K})$  but a marginalized distribution  $p(z'|\mathbf{y}, \mathbf{z}, \mathbf{y}')$  by jointly estimating parameters. In this case, the target density is not  $p(\beta, K|\mathbf{z}, \mathbf{y})$  but  $p(\beta, K, z'|\mathbf{z}, \mathbf{y}, \mathbf{y}')$ . Then each unknown parameter from the marginalized density is updated according to the modified acceptance probability

$$\mathcal{A} = \min \left\{ 1, \frac{p(z', \mathbf{z}|\mathbf{y}, \mathbf{y}_i, \hat{\beta}, \hat{K})p(\hat{\beta})p(\hat{K})q(z', \beta, K)}{p(z', \mathbf{z}|\mathbf{y}, \mathbf{y}_i, \beta, K)p(\beta)p(K)q(z', \hat{\beta}, \hat{K})} \right\}. \quad (3)$$

In this paper, we use the second approach to infer the parameters and classify the data for MCMC simulation for comparison since the joint estimation to obtain the marginalized distribution considers the uncertainty even in the classification of the new dataset. We simply design  $q(\hat{z}', \hat{\beta}, \hat{K}) = q(\hat{z}')q(\hat{\beta})q(\hat{K})$  and each proposal distribution is defined by

$$\begin{aligned} q(\hat{z}') &= p(\hat{z}'|\mathcal{Y}, \hat{\beta}, \hat{K}) = \frac{p(\hat{z}', \mathbf{z}|\mathbf{y}, \mathbf{y}_i, \hat{\beta}, \hat{K})}{\sum_{c \in \mathbf{C}} p(\hat{z}' = c, \mathbf{z}|\mathbf{y}, \mathbf{y}_i, \hat{\beta}, \hat{K})} \\ q(\hat{\beta}) &= \mathcal{N}(\hat{\beta}; \beta, 0.1) \\ q(\hat{K}) &= p(\hat{K}) = \frac{1}{K_{\max}} \end{aligned} \quad (4)$$

where we set  $\beta_a = 2$  and  $\beta_b = 10$  for the Gamma distribution. Given this particular setting of the proposal distribution, we obtain the simplified acceptance probability

$$\mathcal{A} = \min \left\{ 1, \frac{\sum_{c \in \mathbf{C}} p(\hat{z}' = c, \mathbf{z}|\mathbf{y}, \mathbf{y}_i, \hat{\beta}, \hat{K}) p(\hat{\beta}) q(\beta)}{\sum_{s \in \mathbf{C}} p(\hat{z}' = s, \mathbf{z}|\mathbf{y}, \mathbf{y}_i, \beta, K) p(\beta) q(\hat{\beta})} \right\}. \quad (5)$$

### 3.3. Integrated Nested Laplace Approximation (INLA)

Suppose that we have a set of hidden variables  $\mathbf{f}$  and a set of observations  $\mathcal{Y}$ , respectively. MCMC can of course be used to infer the marginal density  $p(\mathbf{f}|\mathbf{y}) = \int p(\mathbf{f}, \theta|\mathbf{y}) d\theta$  where  $\theta$  is a set of control parameters. In order to efficiently build the target density, we apply a remarkably fast and accurate functional approximation based on the Integrated Nested Laplace Approximation (INLA) developed by [21]. This algorithm approximates the marginal posterior  $p(\mathbf{f}|\mathcal{Y})$  by

$$\begin{aligned} p(\mathbf{f}|\mathcal{Y}) &= \int p(\mathbf{f}|\mathcal{Y}, \theta) p(\theta|\mathcal{Y}) d\theta \\ &\approx \int \tilde{p}(\mathbf{f}|\mathcal{Y}, \theta) \tilde{p}(\theta|\mathcal{Y}) d\theta \\ &\approx \sum_{\theta_i} \tilde{p}(\mathbf{f}|\mathcal{Y}, \theta) \tilde{p}(\theta|\mathcal{Y}) \Delta_{\theta_i} \end{aligned} \quad (6)$$

where

$$\tilde{p}(\theta|\mathcal{Y}) \propto \frac{p(\mathbf{f}, \mathcal{Y}, \theta)}{p_F(\mathbf{f}|\mathcal{Y}, \theta)} \Big|_{\mathbf{f}=\mathbf{f}^*(\theta)} = \frac{p(\mathcal{Y}|\mathbf{f}, \theta) p(\mathbf{f}|\theta) p(\theta)}{p_F(\mathbf{f}|\mathcal{Y}, \theta)} \Big|_{\mathbf{f}=\mathbf{f}^*(\theta)}. \quad (7)$$

Here,  $F$  denotes a simple functional approximation close to  $p(\mathbf{f}|\mathcal{Y}, \theta)$  such as a Gaussian approximation and  $\mathbf{f}^*(\theta)$  is a value of the functional approximation. For the simple Gaussian approximation case, the proper choice of  $\mathbf{f}^*(\theta)$  is the mode of the Gaussian approximation of  $p_G(\mathbf{f}|\mathcal{Y}, \theta)$ . Given the log of the posterior, we can calculate the mode  $\theta^*$  and its Hessian matrix  $\mathbf{H}_\theta^*$  via Quasi-Newton style optimization by  $\theta^* = \arg_\theta \max \log \tilde{p}(\theta|\mathcal{Y})$  and  $\mathbf{H}_\theta^*$ . Finally we do a grid search from the mode in all directions until  $\log \tilde{p}(\theta^*|\mathcal{Y}) - \log \tilde{p}(\theta|\mathcal{Y}) > \varphi$ , for a given threshold  $\varphi$ .

#### 4. Proposed Approach

Our proposed algorithm estimates the underlying densities for the number of neighbours of probabilistic kNN classification by using Eq. (7). To distinguish it from other model selection approaches, we term this approach *KOREA*, which is an acronym for "**K-OR**der **E**stimation **A**lgorithm" in a Bayesian framework.

##### 4.1. Obtaining the optimal number of neighbours $K^*$

Let  $\mathcal{Y}$  denote a set of observations and let  $\mathbf{f}_K$  be a set of the model parameters given a model order  $K$ . The first step of *KOREA* is to estimate the optimal number of neighbours,  $K^*$ :

$$K^* = \arg_K \max p(K|\mathcal{Y}). \quad (8)$$

According to Eq. (7), we can obtain an approximated marginal posterior distribution by

$$\tilde{p}(K|\mathcal{Y}) \propto \frac{p(\mathcal{Y}, \mathbf{f}_K, K)}{p_F(\mathbf{f}_K|\mathcal{Y}, K)} \Big|_{\mathbf{f}_K(K)=\mathbf{f}_K^*(K)}. \quad (9)$$

This equation has the property that  $K$  is an integer variable while  $\theta$  of Eq. (7) is in general a vector of continuous variables. By ignoring the difference, we can still use the Quasi-Newton method to efficiently obtain optimal  $K^*$ . Alternatively, we can also calculate some potential candidates between 1 and  $K_{\max}$  if  $K_{\max}$  is not too large. Otherwise, we may still use the *Quasi-Newton* style algorithm with a rounding operator which transforms a real value to an integer for  $K$ .

##### 4.2. Bayesian Model Selection for PKNN classification

In general, one of the most significant problems in classification is to infer the joint posterior distribution of  $L$  different hidden classes for  $L$  different observations such that  $\mathbf{z}_{1:L}'^* = \arg_{\mathbf{z}_{1:L}'} \max p(\mathbf{z}_{1:L}'|\mathbf{y}, \mathbf{z}, \mathbf{y}_{1:L}')$ . However, jointly inferring



the hidden variables is not straightforward therefore we make the assumption that the hidden class of the  $i$ -th observation  $z'_i$  is independent to one of the  $j$ -th observation given the  $i$ -th observation  $y'_i$  where  $i \neq j$  and then we have the following simpler form (similar to Naive Bayes):

$$p(\mathbf{z}'_{1:L}|\mathbf{y}, \mathbf{z}, \mathbf{y}'_{1:L}) = \prod_{i=1}^L p(z'_i|\mathbf{y}, \mathbf{z}, \mathbf{y}'_i) \quad (10)$$

where  $p(z'_i|\mathbf{y}, \mathbf{z}, \mathbf{y}'_i)$  is estimated by Eq. (11).

#### 4.2.1. PKNN via KOREA

In the probabilistic kNN model (PKNN), let us define the new dataset with  $L$  data by  $\mathbf{y}'_{1:L}$ , which is not labeled yet. The unknown labels are denoted by  $\mathbf{z}'_{1:L}$ . Here we use  $\mathbf{y}'_i$  and  $z'_i$  for the  $i$ th new observation and its hidden label. That is, we have a hidden variable  $\mathbf{f}_K = z'_i$  of interest given  $\mathbf{z} = \mathbf{z}_{1:N}$ ,  $\mathbf{y} = \mathbf{y}_{1:N}$  and  $\mathbf{y}'_i$  such that  $\mathcal{Y} = (\mathbf{z}, \mathbf{y}, \mathbf{y}'_i)$ . The target posterior is obtained in a similar form to Eq. (9) as

$$\begin{aligned} p(z'_i|\mathbf{y}, \mathbf{z}, \mathbf{y}'_i) &= p(z'_i|\mathcal{Y}) = \int_{K, \beta} p(z'_i, \beta, K|\mathcal{Y}) d\beta dK \\ &= \int p(z'_i|\beta, \mathcal{Y}, K) p(\beta|\mathcal{Y}, K) p(K|\mathcal{Y}) d\beta dK \\ &\approx \sum_{\beta^{(m)}} \sum_{j=1}^{K_{\max}} \left[ p(z'_i|\beta^{(i)}, \mathcal{Y}, K=j) p(\beta^{(m)}|\mathcal{Y}, K=j) \right. \\ &\quad \times p(K=j|\mathcal{Y}) \Delta_{\beta^{(m)}} \left. \right] \\ &\approx \sum_{\beta^{(m)}} \sum_{j=1}^{K_{\max}} \left[ p(z'_i|\beta^{(i)}, \mathcal{Y}, K=j) \tilde{p}(\beta^{(m)}|\mathcal{Y}, K=j) \right. \\ &\quad \times \tilde{p}(K=j|\mathcal{Y}) \Delta_{\beta^{(m)}} \left. \right] \\ &= \sum_{\beta^{(m)}} \sum_{j=1}^{K_{\max}} \lambda_j^{(m)} p(z'_i|\beta^{(m)}, \mathcal{Y}, K=j) \end{aligned} \quad (11)$$

where

$$\lambda_j^{(m)} = \frac{\tilde{p}(\beta^{(m)}|\mathcal{Y}, K=j) \tilde{p}(K=j|\mathcal{Y}) \Delta_{\beta^{(m)}}}{\sum_{\beta^{(a)}} \sum_{b=1}^{K_{\max}} \tilde{p}(\beta^{(a)}|\mathcal{Y}, K=b) \tilde{p}(K=b|\mathcal{Y}) \Delta_{\beta^{(a)}}}. \quad (12)$$

Now we need to know three distributions in the above equation.

1.  $p(z'_i|\beta^{(m)}, \mathcal{Y}, K = i)$ : conditional likelihood
2.  $\tilde{p}(\beta^{(m)}|\mathcal{Y}, K = i)$ : posterior of  $\beta$
3.  $p(K|\mathcal{Y})$ : posterior of  $K$

The first equation among the three above is the conditional distribution and it is defined by

$$p(z'_i|\beta^{(m)}, \mathcal{Y}, K = j) = \frac{p(\mathbf{z}, z'_i|\beta^{(m)}, \mathcal{Y}, K = j)}{\sum_{c \in \mathbf{C}} p(\mathbf{z}, z'_i = c|\beta^{(m)}, \mathcal{Y}, K = j)}. \quad (13)$$

This is a likelihood function given the neighbouring structure. That is,  $p(\mathbf{z}, z'_i|\beta^{(m)}, \mathcal{Y}, K = j)$  explains the fitness between the assumed/given labels  $(\mathbf{z}, z'_i)$  and the given full data  $(\mathbf{y}, \mathbf{y}'_i)$

Another equation is  $\tilde{p}(\beta^{(m)}|\mathcal{Y}, K = j)$  but we defer the estimation of this distribution since it can be automatically estimated when we estimate the last distribution  $p(K|\mathcal{Y})$ . Therefore, we infer the last equation first. The last equation is the marginal posterior of  $K$  and using a similar approach to INLA it is defined by

$$\begin{aligned} \tilde{p}(K|\mathcal{Y}) &\propto \frac{p(\mathbf{z}, \beta, K|\mathbf{y}, \mathbf{y}'_i)}{p_G(\beta|\mathbf{z}, \mathbf{y}, \mathbf{y}'_i, K)} \Big|_{\beta=\beta^*(K)} = \frac{\sum_{c \in \mathbf{C}} p(z'_i, \mathbf{z}, \beta, K|\mathbf{y}, \mathbf{y}'_i)}{p_G(\beta|\mathbf{z}, \mathbf{y}, \mathbf{y}'_i, K)} \Big|_{\beta=\beta^*(K)} \\ &= \frac{p(\beta)p(K) \sum_{c \in \mathbf{C}} p(z'_i = c, \mathbf{z}|\beta, K, \mathbf{y}, \mathbf{y}'_i)}{p_G(\beta|\mathbf{z}, \mathbf{y}, \mathbf{y}'_i, K)} \Big|_{\beta=\beta^*(K)}. \end{aligned} \quad (14)$$

As we can see the denominator is the approximation of the second distribution of interest so we can reuse it i.e.  $\tilde{p}(\beta|\mathcal{Y}, K) = p_G(\beta|\mathbf{z}, \mathbf{y}, \mathbf{y}'_i, K)$  which is a Gaussian approximation of  $p(\beta|\mathcal{Y}, K) \propto p(\mathbf{z}|\mathbf{y}, \mathbf{y}'_i, K)p(\beta) = \sum_{c \in \mathbf{C}} p(z'_i = c, \mathbf{z}|\mathbf{y}, \mathbf{y}'_i, K)p(\beta)$ .

We also easily obtain the marginal posterior of  $\beta$  which is  $p(\beta|\mathcal{Y})$ . Since the marginal posterior is approximated by  $p(\beta|\mathcal{Y}) \approx \tilde{p}(\beta|\mathcal{Y}) = \sum_{j=1}^{K_{\max}} \tilde{p}(\beta|\mathcal{Y}, K = j)\tilde{p}(K = j|\mathcal{Y})$ , we can simply reconstruct the distribution by reusing the previously estimated distributions. When we have  $\mu_\beta^{(j)} = \mathbf{E}(\beta|\mathcal{Y}, K = j)$  and  $\sigma_\beta^{(j)2} = \mathbf{V}(\beta|\mathcal{Y}, K = j)$ , then we have

$$\mu_\beta = \sum_{j=1}^{K_{\max}} \tilde{\alpha}_j \mu_\beta^{(j)} \text{ and } \sigma_\beta^2 = \sum_{j=1}^{K_{\max}} \tilde{\alpha}_j \left[ \sigma_\beta^{(j)2} + \left\{ \mu_\beta - \mu_\beta^{(j)} \right\}^2 \right]. \quad (15)$$

Finally, we can obtain the target distribution of interest  $p(z'_i|\mathbf{y}, \mathbf{z}, \mathbf{y}'_i)$  with three distributions. Since we can now estimate the target distribution as a mixture distribution, we can also obtain the expectation and variance as follows:

$$\begin{aligned}\mathbf{E}(z'_i|\mathcal{Y}) &= \sum_{\beta^{(m)}} \sum_{j=1}^{K_{\max}} \lambda_j^{(m)} \mu_{m,j}^{(i)} \\ \mathbf{V}(z'_i|\mathcal{Y}) &= \sum_{\beta^{(m)}} \sum_{j=1}^{K_{\max}} \lambda_j^{(m)} \left[ \Sigma_{m,j}^{(i)} + \left\{ \mathbf{E}(z'_i) - \mu_{m,j}^{(i)} \right\}^2 \right]\end{aligned}\quad (16)$$

where  $\mu_{m,j}^{(i)} = \mathbf{E}(z'_i|\mathcal{Y}, \beta^{(m)}, K = j)$  and  $\Sigma_{m,j}^{(i)} = \mathbf{V}(z'_i|\mathcal{Y}, \beta^{(m)}, K = j)$ . Here  $p(\beta) = \mathcal{G}(\beta; a_\beta, b_\beta)$  and  $\mathcal{IG}(\cdot; a, b)$  represents inverse Gamma distribution with hyper-parameters  $a$  and  $b$ . In this paper, we set  $a = 2$  and  $b = 10$  yielding an almost flat prior.

### 4.3. Additional Neighbouring Rules

#### 4.3.1. A Boltzmann modelling with equal weights

In the conventional Boltzmann modelling for the neighbouring structure, the interaction rate  $\beta$  is divided by a fixed  $K$  as shown in Eq. (2). This results in each neighbour having its own different weight. Therefore, we need to apply an equal weight to the neighbours by varying  $K$  for the different neighbouring structure. In order to build this strategy, we adopt three sequential approaches: (i) obtain a neighbour structure in the same way as conventional Boltzmann modelling; (ii) modify the structure by transforming from a directed graph to an undirected graph. If  $j \in ne(i)$  but  $i \notin ne(j)$  then we add  $i$  into  $ne(j)$  for  $i \neq j$ ; and (iii) apply the pseudo likelihood for the likelihood. In this paper, we name this modelling as *Boltzmann*<sup>(2)</sup> modelling.

## 5. Simulation Results

The performance of our algorithm is tested with a collection of benchmark datasets. All of the datasets (test and training) used in this paper can be found at <http://mathsci.ucd.ie/~nial/dnn/>. The six well-known benchmark datasets are presented in Table 1. We test the performance by using 4-fold cross validation for a fair comparison with all approaches although our proposed approach does not require it due to the Bayesian nature of it.

---

**Algorithm 1** PKNN classifier via *KOREA*

---

**Require:** Given  $N$  observations,  $(\mathbf{y}, \mathbf{z}) = (\mathbf{y}_{1:N}, \mathbf{z}_{1:N})$ , a new testing set with  $L$  observations,  $\mathbf{y}' = \mathbf{y}'_{1:L}$  and a set of classes  $\mathbf{C}$

1: **for**  $i = 1$  to  $L$  **do**

2: Obtain a new observation  $\mathbf{y}'_i$  and set  $\mathcal{Y} = (\mathbf{y}, \mathbf{z}, \mathbf{y}'_i, \hat{\beta})$ .

- **Calculate**  $\tilde{p}(K|\mathcal{Y}, \hat{\beta})$ .

3: **for**  $j = 1$  to  $K_{\max}$  **do**

4: Calculate the approximate conditional posterior  $\tilde{p}(\beta|\mathcal{Y}, K = j) = p_G(\beta|\mathcal{Y}, K = j)$  by using Gaussian approximation of  $p(\beta|\mathcal{Y}, K = j) \propto p(\mathbf{z}|\mathbf{y}, \mathbf{y}'_i, \beta, K = j)p(\beta)$ .

5: Obtain  $\mu_{\beta}^{(j)} = \mathbf{E}(\beta|\mathcal{Y}, K = j)$  and  $\sigma_{\beta}^{(j)2} = \mathbf{V}(\beta|\mathcal{Y}, K = j)$ .

6: Calculate an unnormalized posterior for  $K = j$ ,  $\alpha_j = \tilde{p}(K = j|\mathcal{Y}) \propto \frac{p(\mu_{\beta}^{(j)})p(K=j)\sum_{c \in \mathbf{C}} p(z'_i=c, \mathbf{z}|\mu_{\beta}^{(j)}, K=j, \mathbf{y}, \mathbf{y}'_i)}{p_G(\mu_{\beta}^{(j)}|\mathcal{Y}, K=j)}$ .

7: **end for**

8: Normalize the model order weights by  $\tilde{\alpha}_s = \frac{\alpha_s}{\sum_{j=1}^{K_{\max}} \alpha_j}$  for all  $s \in \{1, 2, \dots, K_{\max}\}$ .

9: Calculate the mean  $\mu_{\beta}$  and variance  $\sigma_{\beta}^2$  of marginal posterior of  $\beta$  from Eq. (15).

10:  $\mathbf{S}_{\beta} = \{\beta|0 < \beta = \mu_{\beta} \pm i\sigma_{\beta} < \beta_{\max} \text{ for } i = 1, 2, \dots\}$ .

11: Calculate an unnormalize weight  $\lambda_j^{(m)} = \tilde{p}(\beta^{(m)}|\mathcal{Y}, K = j)\alpha_j$  for  $j = 1, 2, \dots, K$  and  $m = 1, 2, \dots, |\mathbf{S}_{\beta}|$ .

12: Obtain  $\tilde{\lambda}_j^{(m)} = \frac{\lambda_j^{(m)}}{\sum_{n=1}^{|\mathbf{S}_{\beta}|} \sum_{k=1}^{K_{\max}} \lambda_k^{(n)}}$  for all  $j \in \{1, 2, \dots, K_{\max}\}$  and all  $m \in \{1, 2, \dots, |\mathbf{S}_{\beta}|\}$  from Eq. (12).

- **Calculate the solution of**  $p(z'_i|\mathcal{Y})$ .

13: **for**  $m = 1$  to  $|\mathbf{S}_{\beta}|$  **do**

14: **for**  $j = 1$  to  $K_{\max}$  **do**

15: **for**  $c \in \mathbf{C}$  **Get**  $\tau_{j,c} = p(z'_i = c, \mathbf{z}|\mathbf{y}, \mathbf{y}'_i, K = j, \beta^{(m)})$ .

16: **for**  $c \in \mathbf{C}$  **Get**  $\tau_{j,c}^{(m)} = \frac{\tau_{j,c}}{\sum_{l \in \mathbf{C}} \tau_{j,l}}$ .

17: **end for**

18: **end for**

19: Calculate  $p(z'_i = c|\mathcal{Y}) = \sum_{m=1}^{|\mathbf{S}_{\beta}|} \sum_{k=1}^{K_{\max}} \tau_{k,c}^{(m)} \tilde{\lambda}_k^{(m)}$  for all  $c \in \mathbf{C}$ .

20: Calculate the expectation and variance of  $z_i$  from Eq. (16).

21: **end for**

---

Table 1: Benchmark datasets:  $C$  (the number classes),  $d$  (the dimension of the data),  $N_{total}$  (the total number of data)

Name of data	$C$	$d$	$N_{total}$
Crabs	4	5	200
Fglass	4	9	214
Meat	5	1050	231
Oliveoil	3	1050	65
Wine	3	13	178
Iris	3	4	150

Figure 2 demonstrates reconstructed densities of a testing datum. While top subgraphs show the 2 dimensional densities  $p(\beta, K|\mathcal{Y})$ , bottom sub-figures represent the 1 dimensional densities  $p(K|\mathcal{Y})$  for all datasets. The graphs illustrate that the distribution is not unimodal but a complex multi-modal distribution. This also suggests that selecting an appropriate number of neighbours for PKNN is critical to obtain high accuracy.

Asymptotically, MCMC with a large number of iterations will converge and therefore can be used in principle to estimate the underlying posterior density. Thus, we can check whether the reconstructed density using KOREA is close to that estimated by MCMC with a very large number of iterations in order to validate the our proposed algorithm. Two subgraphs of figure 3 visualize the similarity between reconstructed posterior densities of a testing data of wine dataset by KOREA (red circle line) and MCMC (blue cross line) with small (top) and large (bottom) number of samples. (For MCMC, we set the sample size by 100 for small size and 10000 for large size respectively.) As we can see in the figures, our proposed algorithm KOREA is closely approximated to the MCMC algorithm with a large number of iterations size which is commonly regarded as underlying reference or pseudo-ground truth density. In order to measure the similarity between the reconstructed densities by MCMC and KOREA, we use four different metrics as shown in figure 4: Root Mean Square Error (RMSE), Peak Signal-to-Noise Ratio (PSNR), Kullback Leibler Distance (KLD) and Structural SIMilarity (SSIM) [22]. As in the case of figure 3, MCMC with a large sample size produces densities very close to those produced by our proposed KOREA algorithm. As the number of MCMC samples increases, RMSE and KLD decrease while PSNR and SSIM increases for all datasets.

Table 2 demonstrates the performance of the each algorithms based on F-measure for four cases: kNN, PKNN, KOREA (average) and KOREA (optimal).

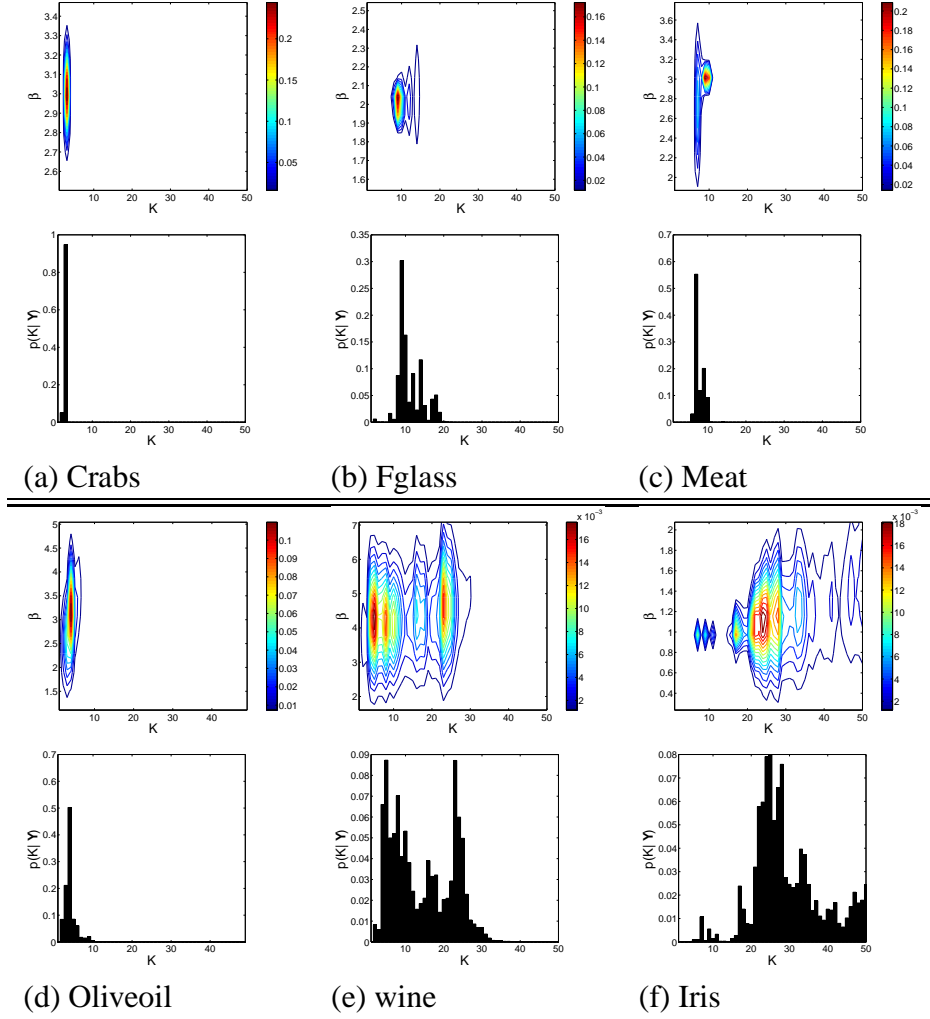
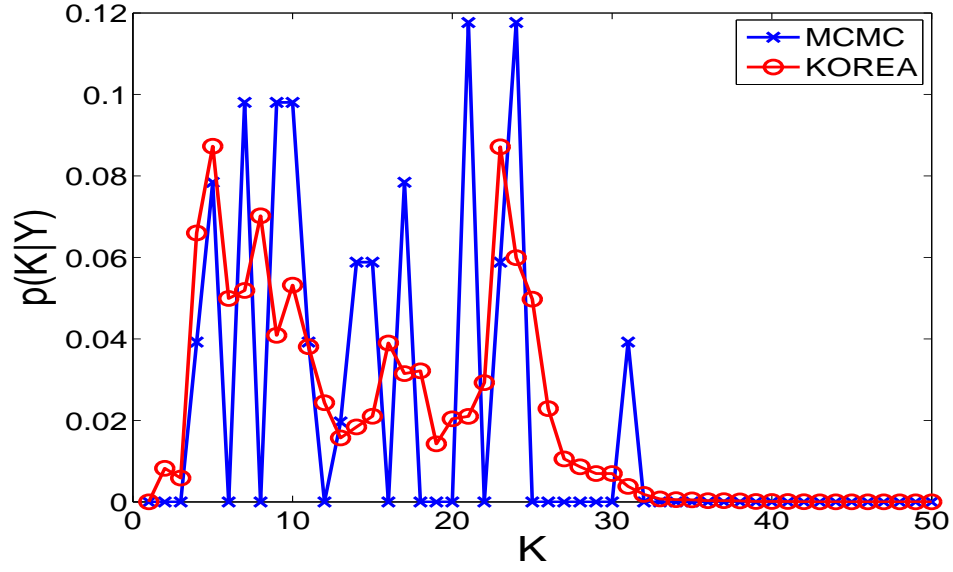


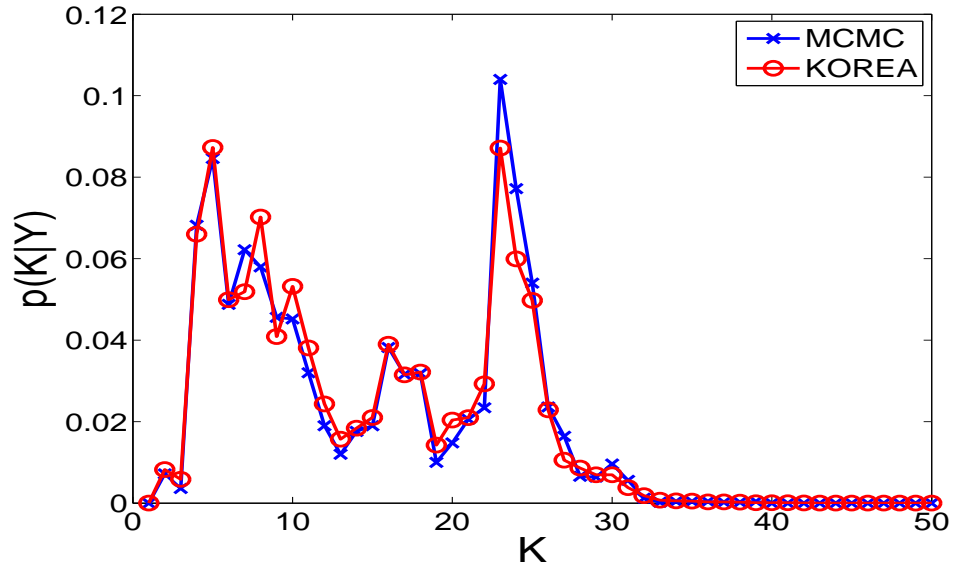
Figure 2: Posterior distribution  $p(K, \beta | \mathcal{Y})$  [top] and its marginalized posterior density  $p(K | \mathcal{Y})$  [bottom] via KOREA

Since MCMC produces results which are very close to that of KOREA as shown in figures 3 and 4, we did not present these results. KOREA (average) and KOREA (optimal) represent the mean (marginalized) estimate and MAP estimate of KOREA, respectively. As we can see in the table, KOREA works superior to other conventional approaches for all datasets. The results with the best performance are highlighted in bold in this table.

In addition, we compared the simulation times for each of the algorithms. Ta-



(a)  $N_{MCMC} = 100$



(b)  $N_{MCMC} = 10000$

Figure 3: Comparison between MCMC and KOREA for wine dataset,  $N_{MCMC}$  denotes the number of MCMC iterations.

ble 3 demonstrates the execution time for all algorithms. Our proposed algorithm (PKNN with KOREA) is slower than conventional kNN and PKNN with fixed  $K$

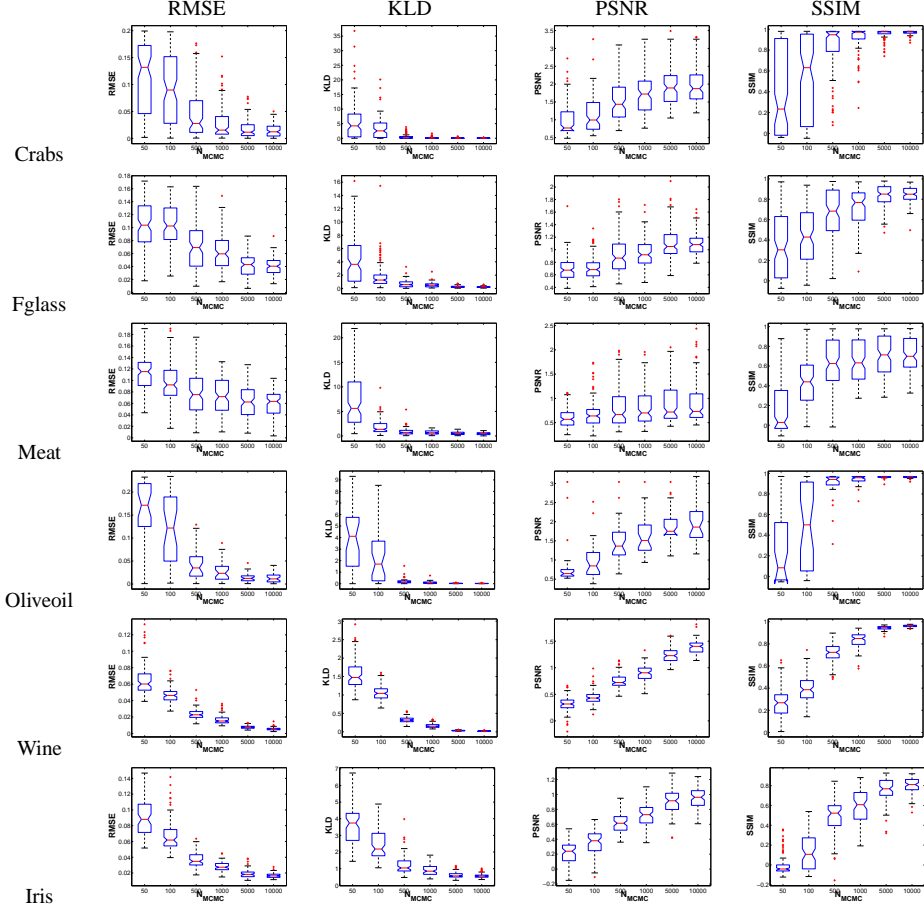


Figure 4: Implicit similarity check between the reconstructed densities by MCMC and KOREA via four well-known metrics.

but it is much faster than MCMC technique which is regarded as one of the best approaches to infer the model parameters and number of neighbours in Bayesian framework. From the point of the accuracy of table 2 and the execution time of table 3, we eventually find that PKNN can be efficiently improved by using our proposed KOREA algorithm and this is a very practically useful technique compared to the conventional approaches including KNN, PKNN and MCMC.

## 6. Discussion

Our proposed algorithm uses an approach similar to the idea of INLA by replacing the model parameters with the model order (the number of neighbours,  $k$ ).



Table 2: Comparison of F-measures with varying neighbouring structures. The results with the best performance are written in bold.

Methods	Data	Asymmetric model	Symmetric Boltzman	Boltzman <sup>(2)</sup>
KNN	Crabs	0.72±0.08	0.74±0.08	0.74±0.08
	Fglass	0.64±0.06	0.67±0.05	0.67±0.05
	Meat	0.68±0.07	0.70±0.07	0.70±0.07
	Oliveoil	0.74±0.12	0.71±0.10	0.71±0.10
	Wine	0.97±0.01	0.97±0.01	0.97±0.01
	Iris	0.57±0.10	0.55±0.10	0.55±0.10
PKNN	Crabs	0.75±0.09	0.75±0.09	0.75±0.08
	Fglass	0.73±0.06	0.74±0.06	0.69±0.06
	Meat	0.70±0.07	0.71±0.07	0.70±0.06
	Oliveoil	0.72±0.11	0.73±0.11	0.70±0.11
	Wine	0.98±0.01	0.98±0.01	0.98±0.02
	Iris	0.57±0.12	0.57±0.12	0.53±0.10
KOREA (average)	Crabs	0.86±0.11	<b>0.89±0.11</b>	0.87±0.09
	Fglass	0.76±0.09	0.77±0.07	<b>0.81±0.08</b>
	Meat	0.68±0.12	<b>0.75±0.06</b>	0.71±0.06
	Oliveoil	<b>0.82±0.17</b>	0.76±0.19	0.73±0.20
	Wine	0.99±0.12	<b>0.99±0.02</b>	0.98±0.02
	Iris	<b>0.62±0.15</b>	0.58±0.17	0.56±0.03
KOREA (optimal)	Crabs	0.86±0.13	<b>0.89±0.11</b>	0.87±0.09
	Fglass	0.79±0.04	0.76±0.08	0.79±0.07
	Meat	0.70±0.11	0.73±0.07	0.69±0.13
	Oliveoil	0.80±0.17	0.78±0.17	0.76±0.19
	Wine	<b>0.99±0.02</b>	<b>0.99±0.02</b>	0.98±0.02
	Iris	0.57±0.17	0.56±0.19	0.48±0.16

This means that we can speed up the computation by embedding (Quasi-)Newton method for Laplace approximation rather than grid sampling as done in the original INLA. However, as we can see in Fig. 2, the posterior is not unimodal so we can find local optima rather than global optima for the maximal mode of the posterior if we use such a simple Laplace approximation. Therefore, instead of (Quasi-)Newton methods employed in the original INLA, we reconstructed the density with relatively slower grid approach for the real datasets in the PKNN

Table 3: Time comparison: the average of the execution times

Data	KNN	PKNN	MCMC (10000 runs)	KOREA
Crabs	0.10	0.46	168.76	9.77
Fglass	0.11	0.52	200.59	10.61
Meat	0.12	0.92	270.46	15.66
Oliveoil	0.02	0.13	34.58	1.90
Wine	0.08	0.30	129.03	6.25
Iris	0.07	0.26	95.47	5.14

of this paper. Of course, if the distribution is uni-modal, then we can use the Quasi-Newton method to speed up the algorithm.

## 7. Conclusion

We proposed a model selection algorithm for probabilistic k-nearest neighbour (PKNN) classification which is based on functional approximation in Bayesian framework. This algorithm has several advantages compared to other conventional model selection techniques. First of all, the proposed approach can quickly provide a proper distribution of the model order  $k$  which is not given by other approaches, in contrast to time consuming techniques like MCMC. In addition, since the proposed algorithm is based on a Bayesian scheme, we do not need to run cross validation which is usually used for the performance evaluation. The proposed algorithm can also inherit the power of the fast functional approximation of INLA. For instance, it can quickly find the optimal number of neighbours  $k$  and efficiently generate the grid samples by embedding Quasi-Newton method if the posterior is uni-modal. Lastly, the proposed approach can calculate the model average which is marginalized posterior  $p(\mathbf{x}|\mathcal{Y}) = \int_{\mathcal{M}} p(\mathbf{x}|\mathcal{Y}, \mathcal{M})p(\mathcal{M}|\mathcal{Y})d\mathcal{M}$ . We also remark that our algorithm is based on a pseudo-likelihood approximation of the likelihood and suggest that, although our algorithm has yielded good performance, further improvements may result by utilising more accurate approximations of the likelihood, albeit at the expense of computational run time.

### Acknowledgements

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2012- H0301-12-3007) supervised by the NIPA (National IT

Industry Promotion Agency). Nial Friel's research was supported by a Science Foundation Ireland Research Frontiers Program grant, 09/RFP/MTH2199.

## References

- [1] C. C. Holmes, N. M. Adams, A probabilistic nearest neighbour method for statistical pattern recognition, *Journal of the Royal Statistical Society - Series B: Statistical Methodology* 64 (2) (2002) 295–306.  
URL <http://doi.wiley.com/10.1111/1467-9868.00338>
- [2] L. Cucala, J.-M. Marin, C. P. Robert, D. M. Titterton, A bayesian re-assessment of nearest-neighbor classification, *Journal of the American Statistical Association* 104 (2009) 263–273.
- [3] S. Manocha, M. Girolami, An empirical analysis of the probabilistic k-nearest neighbour classifier, *Pattern Recognition Letters* 28 (13) (2007) 1818–1824.  
URL <http://eprints.gla.ac.uk/13913/>
- [4] N. Friel, A. N. Pettitt, Classification via distance nearest neighbours, *Statistics and Computing* 21 (2011) 431–437.
- [5] H. Akaike, A new look at the statistical model identification, *IEEE Transactions on Automatic Control* 19 (1974) 716–723.
- [6] G. Schwarz, Estimating the dimension of a model, *The Annals of Statistics* 6 (2) (1978) 461–464.
- [7] D. J. Spiegelhalter, N. G. Best, B. P. Carlin, Bayesian measures of model complexity and fit, *Journal of the Royal Statistical Society, Series B* 64 (2002) 583–639.
- [8] J. E. Besag, Spatial interaction and the statistical analysis of lattice systems (with discussion), *Journal of the Royal Statistical Society, Series B* 36 (1974) 192–236.
- [9] I. Murray, Z. Ghahramani, D. MacKay, Mcmc for doubly-intractable distributions, in: *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, AUAI Press, Arlington, Virginia, 2006.
- [10] C. Varin, N. Reid, D. Firth, An overview of composite likelihood methods, *Statistica Sinica* 21 (2011) 5–42.

- [11] U. Grenander, M. I. Miller, Representations of knowledge in complex systems, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 56 (4) (1994) 549–603.
- [12] M. Stephens, Bayesian Analysis of Mixture Models with an Unknown number of components – an alternative to reversible jump methods, *The Annals of Statistics* 28 (1) (2000) 40–74.
- [13] S. Richardson, P. J. Green, On bayesian analysis of mixtures with an unknown number of components, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 59 (4) (1997) 731–792.
- [14] B. P. Carlin, S. Chib, Bayesian model choice via markov chain monte carlo methods, *Journal of the Royal Statistical Society, Series B* 57 (1995) 473–484.
- [15] S. J. Godsill, On the relationship between markov chain monte carlo methods for model uncertainty, *Journal of Computational and Graphical Statistics* 10 (2001) 230–248.
- [16] J. Wang, P. Neskovic, L. N. Cooper, Neighborhood size selection in the k-nearest-neighbor rule using statistical confidence, *Pattern Recognition* 39.
- [17] D. J. Hand, V. Vinciotti, Choosing k for two-class nearest neighbour classifiers with unbalanced classes, *Pattern Recognition Letters* 24 (910) (2003) 1555 – 1562.
- [18] C. Domeniconi, J. Peng, D. Gunopulos, Adaptive metric nearest neighbor classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2000) 1281–1285.
- [19] R. Guo, S. Chakraborty, Bayesian adaptive nearest neighbor, *Statistical Analysis and Data Mining* 3 (2010) 92–105.
- [20] L. Cucala, J.-M. Marin, C. Robert, M. Titterton, A Bayesian Reassessment of Nearest-Neighbor Classification, *Journal of the American Statistical Association* 104 (485) (2008) 263–273.  
URL <http://arxiv.org/abs/0802.1357>
- [21] H. Rue, S. Martino, N. Chopin, Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations, *Journal Of The Royal Statistical Society Series B* 71 (2) (2009) 319–392.

- [22] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: From error visibility to structural similarity, *IEEE Transactions on Image Processing* 13 (4) (2004) 600–612.